IN THE CLAIMS

1. (currently amended) A method for execution on a signal processing unit for constructing a control flow graph ~~(CFG)~~ from a computer executable program the instructions of which belong to one or more computer architecture instruction sets, said method comprising ~~the steps of~~

defining a number of block leader types including at least one type related to an instruction set change, block leaders specifying basic block boundaries in the program, said basic blocks including instructions or data ~~(702)~~,

building a control flow graph ~~CFG~~ structure comprising basic blocks found in the program ~~(708)~~,

adding control flow and addressing information to said ~~CFG~~ control flow graph by propagating through said basic blocks and internals thereof ~~(710)~~.

2. (currently amended) The method of claim 1, further comprising ~~a step wherein~~ compacting said control flow graph ~~CFG is compacted~~ by utilizing a ~~CFG~~ control flow graft optimisation technique ~~(712)~~.

3. (currently amended) The method of claim 2, wherein said optimisation technique is ~~substantially~~ at least one of the following: unreachable eliminator, branch unconditionalizer, simple redundant eliminator, redundant load eliminator, dead code eliminator, register liveness analyser, branch rationalizer.

4. (currently amended) The method of claim 1, wherein ~~the step of~~ said adding control flow and addressing information includes iterating through instructions of a single basic block at a time based on ~~the basis of~~ constant propagation information associated with said block.

5. (currently amended)The method of claim 1, wherein ~~the step of~~ said adding control flow and addressing information ~~(710)~~ includes ~~propagation of~~ propagating basic block emulation results from a block to another.

6. (previously presented) The method of claim 1, wherein said block leader types further include an entry for at least one of the following: data symbols intermixed with instructions, program entry point, exception vector, relocation entry point, relocation target point, successor of a branch instruction, target of a branch instruction, function.

7. (currently amended) The method of claim 1, wherein the ~~CFG~~ control flow graph comprises hierarchical levels of sections, functions and basic blocks.

8. (currently amended) The method of claim 1, further comprising ~~the step of~~ reading data from a binary executable file ~~(704)~~.

9. (currently amended) The method of claim 1, further comprising ~~the step of~~ reconstructing an executable from said ~~CFG (714).~~ control flow graph.

10. (currently amended) The method of claim 1, wherein said program is ~~substantially~~ ARM or THUMB architecture specific.

11. (previously presented) A computer program comprising code for carrying out the steps of claim 1 for at least temporary storage in a computer readable medium.

12. (currently amended) A ~~carrier~~ computer readable medium ~~for storing~~ on which a computer executable program is stored for carrying out the steps of claim 1.

13. (currently amended) A system for constructing a control flow graph ~~(CFG)~~ from a computer executable program, the instructions of which belong to one or more computer architecture instruction sets, said system comprising a processing device ~~means (806)~~ and a memory device ~~means (810)~~ for processing and storing instructions and data, and a data transfer device ~~means (808)~~ for accessing data, said system arranged to define a number of block leader types including at least one type related to

an instruction set change, block leaders specifying basic block boundaries in the program, said basic blocks including instructions or data, said system further arranged to build a control flow graph ~~CFG~~ structure comprising basic blocks found in the program, and to add control flow and addressing information to said control flow graph ~~CFG~~ by propagating through said basic blocks and internals thereof.

14. (currently amended) The system of claim 13, further arranged to compact the control flow graph ~~CFG~~ by utilizing a control flow graph ~~CFG~~ optimisation technique.

15. (new) A system for constructing a control flow graph from a computer executable program, the instructions of which belong to one or more computer architecture instruction sets, said system comprising a processing means and a memory means for processing and storing instructions and data, and a data transfer device for accessing data, said system arranged to define a number of block leader types including at least one type related to an instruction set change, block leaders specifying basic block boundaries in the program, said basic blocks including instructions or data, said system further arranged to build a control flow graph structure comprising basic blocks found in the program, and to add control flow and addressing information to said control flow graph by propagating through said basic blocks and internals thereof.